What is claimed is:

1. A system for providing passive screening of transient messages in a distributed computing environment, comprising:

a network interface passively monitoring a transient packet stream at a network boundary comprising receiving incoming datagrams structured in compliance with a network protocol layer;

a packet receiver reassembling one or more of the incoming datagrams into a segment structured in compliance with a transport protocol layer; and

an antivirus scanner scanning contents of the reassembled segment for a presence of at least one of a computer virus and malware to identify infected message contents.

2. A system according to Claim 1, further comprising:

an incoming queue staging each incoming datagram intermediate to reassembly.

3. A system according to Claim 1, further comprising:

a network protocol-specific decoder decoding the reassembled segment prior to scanning.

4. A system according to Claim 1, wherein the antivirus scanner terminates the transient packet stream if the reassembled segment is not infected with at least one of a computer virus and malware.

5. A system according to Claim 1, wherein the antivirus scanner takes an action if the reassembled segment is infected with at least one of a computer virus and malware.

6. A system according to Claim 5, wherein the action comprises at least one of logging an infection; generating a warning; spoofing a valid datagram in place of the infected datagram; and acquiescing to the infection.

7. A system according to Claim 1, further comprising:

a protocol-specific queue staging each reassembled segment with other reassembled segments sharing the same transport protocol layer.

8. A system according to Claim 7, further comprising:

an information record storing information dependent on the same transport protocol layer with the staged reassembled segment.

9. A system according to Claim 8, further comprising:

a contents record storing the contents with the staged reassembled segment.

10. A system according to Claim 8, wherein the information comprises at least one of a source address, source port number, destination address, destination port number, URL, file name, user name, sender identification, recipient identification, and subject.

11. A system according to Claim 1, further comprising:

a protocol-specific module processing each reassembled datagram based on the transport layer protocol employed by the reassembled datagram.

12. A system according to Claim 11, wherein the transport layer protocol comprises at least one of HTTP, FTP, SMTP, POP3, NNTP, and Gnutella.

13. A system according to Claim 1, further comprising:

an event correlator analyzing the transient packet stream for events indicative of a network service attack.

14. A system according to Claim 13, further comprising:

a data repository maintaining each event.

15. A system according to Claim 1, wherein the distributed computing environment is TCP/IP-compliant and each incoming message is SMTP-compliant.

16. A method for providing passive screening of transient messages in a distributed computing environment, comprising:

passively monitoring a transient packet stream at a network boundary comprising receiving incoming datagrams structured in compliance with a network protocol layer;

reassembling one or more of the incoming datagrams into a segment structured in compliance with a transport protocol layer; and

scanning contents of the reassembled segment for a presence of at least one of a computer virus and malware to identify infected message contents.

17. A method according to Claim 16, further comprising:

staging each incoming datagram intermediate to reassembly.

18. A method according to Claim 16, further comprising:

decoding the reassembled segment prior to scanning.

19. A method according to Claim 16, further comprising:

terminating the transient packet stream if the reassembled segment is not infected with at least one of a computer virus and malware.

20. A method according to Claim 16, further comprising:

taking an action if the reassembled segment is infected with at least one of a computer virus and malware.

21. A method according to Claim 20, further comprising:

executing the action, comprising at least one of:

logging an infection;

generating a warning;

spoofing a valid datagram in place of the infected datagram; and

acquiescing to the infection.

22. A method according to Claim 16, further comprising:

staging each reassembled segment with other reassembled segments sharing the same transport protocol layer.

23. A method according to Claim 22, further comprising:

storing information dependent on the same transport protocol layer with the staged reassembled segment.

24. A method according to Claim 23, further comprising:

storing the contents with the staged reassembled segment.

25. A method according to Claim 23, wherein the information comprises at least one of a source address, source port number, destination address, destination port number, URL, file name, user name, sender identification, recipient identification, and subject.

26. A method according to Claim 16, further comprising:

processing each reassembled datagram based on the transport layer protocol employed by the reassembled datagram.

27. A method according to Claim 26, wherein the transport layer protocol comprises at least one of HTTP, FTP, SMTP, POP3, NNTP, and Gnutella.

28. A method according to Claim 16, further comprising:

analyzing the transient packet stream for events indicative of a network service attack.

29. A method according to Claim 28, further comprising:

maintaining each event in a data repository.

30. A method according to Claim 16, wherein the distributed computing environment is TCP/IP-compliant and each incoming message is SMTP-compliant.

31. A computer-readable storage medium holding code for performing the method according to Claims 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, or 30.

32. A system for passively detecting computer viruses and malware and denial of service-type network attacks in a distributed computing environment, comprising:

a network interface receiving copies of datagrams transiting a boundary of a network domain into an incoming packet queue, each datagram being copied from a packet stream;

a packet receiver reassembling one or more such datagrams from the incoming packet queue into network protocol packets, each staged in a reassembled packet queue;

an antivirus scanner scanning each network protocol packet from the reassembled packet queue to ascertain an infection of at least one of a computer virus and malware; and

an event correlator evaluating events identified from the datagrams in the packet stream to detect a denial of service-type network attack on the network domain.

33. A system according to Claim 32, further comprising:

a parser parsing each reassembled datagram into network protocol-specific information and packet content.

34. A system according to Claim 33, wherein the network protocol-specific information comprises a source address, source port number, destination address, destination port number, and URL for HTTP; a file name and user name for FTP; and a sender identification, recipient identification, and subject for SMTP.

35. A system according to Claim 33, further comprising:

a decoder decoding the packet content prior to performing the operation of scanning.

36. A system according to Claim 32, further comprising:

a log logging an occurrence of at least one of the infection and the network attack.

37. A system according to Claim 32, further comprising:

a warning module generating a warning responsive to an occurrence of at least one of the infection and the network attack.

38. A system according to Claim 32, further comprising:

a spoof module sending a spoofed network protocol packet responsive to an occurrence of at least one of the infection and the network attack.

39. A system according to Claim 32, further comprising:

one or more protocol-specific modules implementing one of HTTP, FTP, SMTP, POP3, NNTP, and Gnutella network protocols.

40. A system according to Claim 32, wherein the distributed computing environment is TCP/IP-compliant, each datagram is IP-compliant, and each network protocol packet is TCP-compliant.

41. A method for passively detecting computer viruses and malware and denial of service-type network attacks in a distributed computing environment, comprising:

receiving copies of datagrams transiting a boundary of a network domain into an incoming packet queue, each datagram being copied from a packet stream;

reassembling one or more such datagrams from the incoming packet queue into network protocol packets, each staged in a reassembled packet queue;

scanning each network protocol packet from the reassembled packet queue to ascertain an infection of at least one of a computer virus and malware; and

evaluating events identified from the datagrams in the packet stream to detect a denial of service-type network attack on the network domain.

42. A method according to Claim 41, further comprising:

parsing each reassembled datagram into network protocol-specific information and packet content.

43. A method according to Claim 42, wherein the network protocol-specific information comprises a source address, source port number, destination address, destination port number, and URL for HTTP; a file name and user name for FTP; and a sender identification, recipient identification, and subject for SMTP.

44. A method according to Claim 42, further comprising:

decoding the packet content prior to performing the operation of scanning.

45. A method according to Claim 41, further comprising:

logging an occurrence of at least one of the infection and the network attack.

46. A method according to Claim 41, further comprising:

generating a warning responsive to an occurrence of at least one of the infection and the network attack.

47. A method according to Claim 41, further comprising:

sending a spoofed network protocol packet responsive to an occurrence of at least one of the infection and the network attack.

48. A method according to Claim 41, further comprising:

implementing at least one of HTTP, FTP, SMTP, POP3, NNTP, and Gnutella network protocols.

49. A method according to Claim 41, wherein the distributed computing environment is TCP/IP-compliant, each datagram is IP-compliant, and each network protocol packet is TCP-compliant.

50. A computer-readable storage medium holding code for performing the method according to Claims 41, 42, 43, 44, 45, 46, 47, 48, or 49.